CS 70 Discrete Mathematics for CS
Fall 2004 Rao

Lecture 9

Error Correcting Codes

Basic Error-Correction

We will consider two situations in which we wish to transmit information on an unreliable channel. The first is exemplified by the internet, where the information (say a file) is broken up into packets, and the unreliability is manifest in the fact that some of the packets are lost during transmission. Suppose that the message consists of n packets and suppose that k packets are lost during transmission. We will show how to encode the initial message consisting of n packets into a redundant encoding consisting of $n+k$ packets such that the recipient can reconstruct the message from any n received packets. Note that in this setting the packets are labelled and thus the recipient knows exactly which packets were dropped during transmission.

Assume that the contents of each packet can be represented as a number modulo q for some prime q. The error-correcting code that we shall use is based on polynomials over the Galois Field $GF(q)$. Denote the message by $m_1,\ldots,m_n$. We will assume that q is sufficiently large so that $q > n+k$. Consider the unique polynomial $P(x)$ of degree $n-1$ such that $P(i) = m_i$ for $1 \le i \le n$. The encoded message is $c_j = P(j)$ for $1 \le j \le n+k$. Using Lagrange Interpolation, the recipient can reconstruct the polynomial $P(x)$ from the value of $P(x)$ at any n points. This error-correcting scheme is therefore optimal—it can recover the n characters of the transmitted message from any n received characters.

The Berlekamp-Welch Algorithm

The second situation that we shall consider is more challenging. In this situation, Alice wishes to send Bob a message over a noisy channel. This time, instead of erasures or packet losses, some of the transmitted packets are corrupted. What makes this situation more challenging for Bob is that he does not know which of the received packets (characters) are corrupted and which were transmitted accurately. Polynomials still provide an optimal scheme for dealing with this situation. Alice must transmit $n+2k$ characters to enable Bob to recover from k general errors, i.e., the encoded message is now $c_j = P(j)$ for $1 \le j \le n+2k$.

The received message $R(j)$ for $1 \le j \le n+2k$ differs from the polynomial $P(x)$ at k points. How can Bob reconstruct $P(x)$ from these $n+2k$ values $R(j)$? Our first observation is that if Bob can find any polynomial $P'(x)$ of degree $n-1$ that agrees with $R(x)$ at $n+k$ points then $P'(x) = P(x)$. This is because out of the $n+k$ points there are at most k errors, and therefore on at least n points $P'(x) = P(x)$. But a polynomial of degree $n-1$ is uniquely defined by its values at n points.

But how do we find such a polynomial? We could try to guess where the k errors lie, but this would take too long (it would take exponential time, in fact). A very clever polynomial-time algorithm for this problem was invented by Berlekamp and Welch. The main idea is to describe the received message $R(x)$ (which because of the errors is not a polynomial) as a ratio of polynomial. Let $e_1,\ldots,e_k$ be the k positions at which errors occurred. Define the error locator polynomial $E(x) = (x-e_1)(x-e_2)\cdots(x-e_k)$. $E(x)$ is 0 at exactly the k points at which errors occurred. Now observe that for all $n+2k$ points $1 \le x \le n+2k$, $P(x)E(x) = R(x)E(x)$.

This is true at points x at which no error occurred since $P(x) = R(x)$ and at points x at which an error occurred since $E(x) = 0$.

Now let $Q(x) = P(x)E(x)$. Then $Q(x)$ is a polynomial of degree $n+k-1$ and is therefore specified by $n+k$ coefficients. $E(x)$ is a polynomial of degree k and is described by $k+1$ coefficients. Note that the coefficient of x^k is 1, so in fact there are only k unknowns here. Moreover we have $n+2k$ linear equations (exercise: write these equations explicitly) $Q(x) = R(x)E(x)$ for $1 \leq x \leq n+2k$. Here the unknowns are the coefficients of the polynomials $Q(x)$ and $E(x)$, and the $R(x)'s$ are the received values and therefore known.

Example. Suppose we want to send the packets "1," "3," and "7." Then we interpolate to find the polynomial

$$P(X) = X^2 + X + 1,$$

which is the unique polynomial of degree 2 such that $P(0) = 1$, $P(1) = 3$, and $P(2) = 7$.

Now we transmit the $n+2k = 5$ messages $P(0) = 1$, $P(1) = 3$, $P(2) = 7$, $P(3) = 13$, and $P(4) = 21$. Suppose $P(1)$ is corrupted, so the receiver receives 0 instead of 3 in that packet.

Let $E(X) = X - e$ be the error-locator polynomial—we don't know what e is yet—and let $R(X)$ be the polynomial whose values at $0, \ldots, 4$ are precisely the values we received over the channel. Then clearly

$$P(X)E(X) = R(X)E(X)$$

for $X = 0, 1, \ldots, 4$ (if the corruption occured at position i, then $E(i) = 0$, so equality trivially holds, and otherwise $P(i) = R(i)$). We don't know what P is (though we do know it is a degree 2 polynomial) and we don't know what E is either, but using the relationship above we can obtain a linear system whose solutions will be the coefficients of P and E.

Let

$$Q(X) = aX^3 + bX^2 + cX + d = P(X)E(X),$$

where a, b, c, d are unknown coefficients (which we will soon try to determine), so

$$aX^3 + bX^2 + cX + d = R(X)E(X) = R(X)(X - e),$$

which we can rewrite as

$$aX^3 + bX^2 + cX + d + R(X)e = R(X)X.$$

Now we substitute $X = 0$, $X = 1$, $\ldots$, $X = 4$ to get five linear equations (recall that $R(i)$ is the value we received for the fifth packet):

$$\begin{aligned}
d + e &= 1 \\
a + b + c + d &= 0 \\
8a + 4b + 2c + d + 7c &= 14 \\
27a + 9b + 3c + d + 13e &= 39 \\
64a + 16b + 4c + d + 21e &= 84.
\end{aligned}$$

We then solve this linear system for a, b, c, d, e, and this gives us the polynomials $Q(X)$ and $E(X)$. We can then find $P(X)$ by computing the quotient $Q(X)/E(X)$, and from P we can obviously recover the original (uncorrupted) values.

Two points need further discussion. How do we know that the $n+2k$ equations are never inconsistent? This is simple. It follows from the fact that the original polynomial $P(x)$ together with the error locator polynomial $E(x)$ gives a solution.

The more interesting question is this: how do we know that the $n+2k$ equations are independent, i.e., how do we know that the solution $Q'(x)$ and $E'(x)$ that we reconstruct satisfy the property that $E'(x)$ divides $Q'(x)$ and that $\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x)$? To see this notice that $Q(x)E'(x) = Q'(x)E(x)$ for $1 \le x \le n+2k$. This holds trivially whenever $E(x)$ or $E'(x)$ is 0, and otherwise it follows from the fact that $\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = R(x)$. But the degree of $Q(x)E'(x)$ and $Q'(x)E(x)$ is n + 2k - 1. Since these two polynomials are equal at $n+2k$ points, it follows that they are the same polynomial, and thus rearranging we get that $\frac{Q'(x)}{E'(x)} = \frac{Q(x)}{E(x)} = P(x)$.